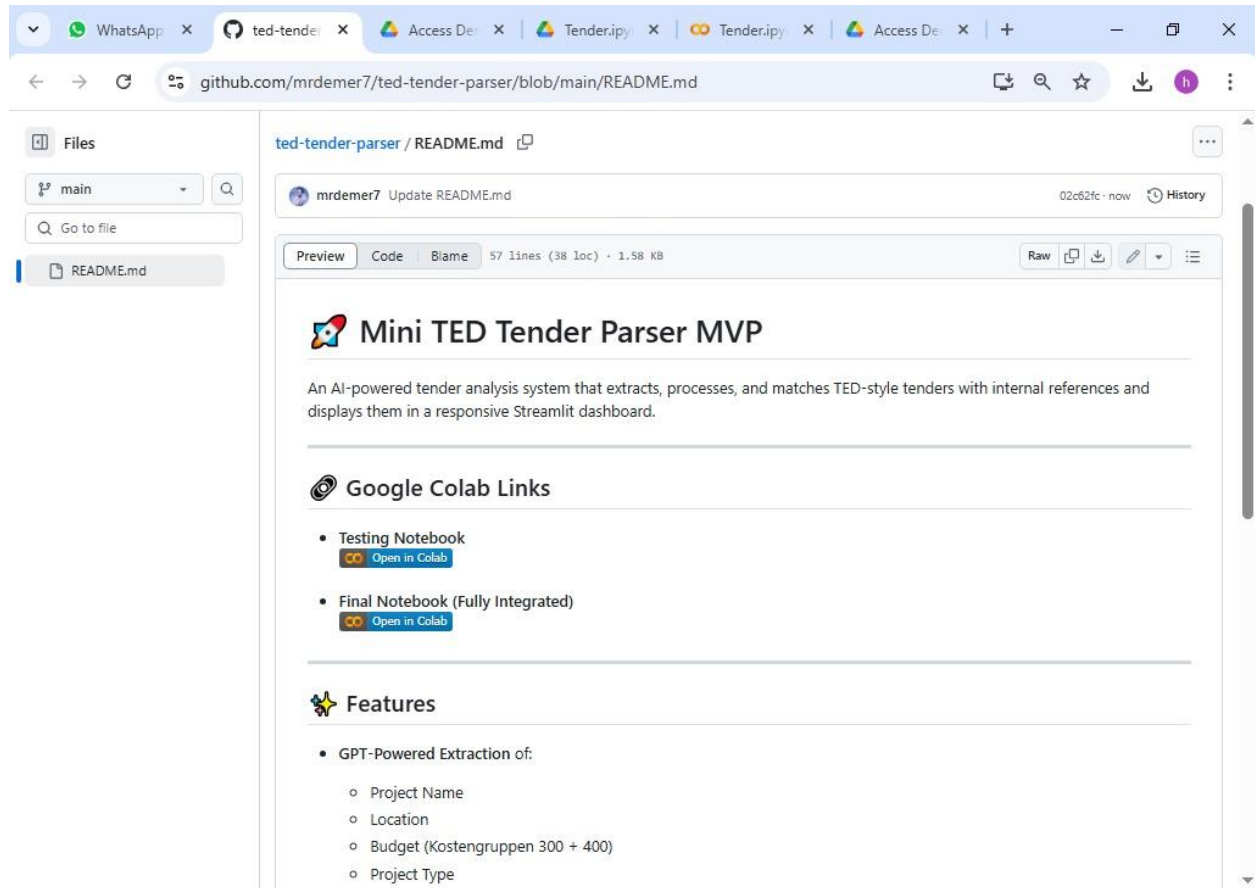


Just click the google cloab link



The screenshot shows a web browser with multiple tabs open, including WhatsApp, ted-tender, Access De, Tender.ipyn, and Access De. The active tab is displaying the GitHub repository page for 'ted-tender-parser' at the path 'blob/main/README.md'. The page title is 'ted-tender-parser / README.md'. The repository owner is 'mrdemer7' and the commit message is 'Update README.md'. The commit hash is '02c62fc' and it was committed 'now'. The file size is '57 lines (38 loc) · 1.58 KB'. The README content includes a title 'Mini TED Tender Parser MVP', a description 'An AI-powered tender analysis system that extracts, processes, and matches TED-style tenders with internal references and displays them in a responsive Streamlit dashboard.', a section 'Google Colab Links' with two links: 'Testing Notebook' and 'Final Notebook (Fully Integrated)', both with 'Open in Colab' buttons, and a section 'Features' with a bullet point 'GPT-Powered Extraction of:' followed by a list of features: 'Project Name', 'Location', 'Budget (Kostengruppen 300 + 400)', and 'Project Type'.

Files

main

Go to file

README.md

ted-tender-parser / README.md

mrdemer7 Update README.md 02c62fc · now History

Preview Code Blame 57 lines (38 loc) · 1.58 KB Raw Copy Download Edit

## Mini TED Tender Parser MVP

An AI-powered tender analysis system that extracts, processes, and matches TED-style tenders with internal references and displays them in a responsive Streamlit dashboard.

### Google Colab Links

- Testing Notebook [Open in Colab](#)
- Final Notebook (Fully Integrated) [Open in Colab](#)


### Features

- GPT-Powered Extraction of:
  - Project Name
  - Location
  - Budget (Kostengruppen 300 + 400)
  - Project Type

# I will give you access

WhatsApp × | ted-tende × | Access De × | Tender.ip × | Tender.ip × | Access De × | +

← → ↻ drive.google.com/file/u/0/d/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF/edit?pli=1 ☆ ⬇ h ⋮


 Google Drive

## You need access


Request access, or switch to an account with access.  
[Learn more](#)

Message (optional)

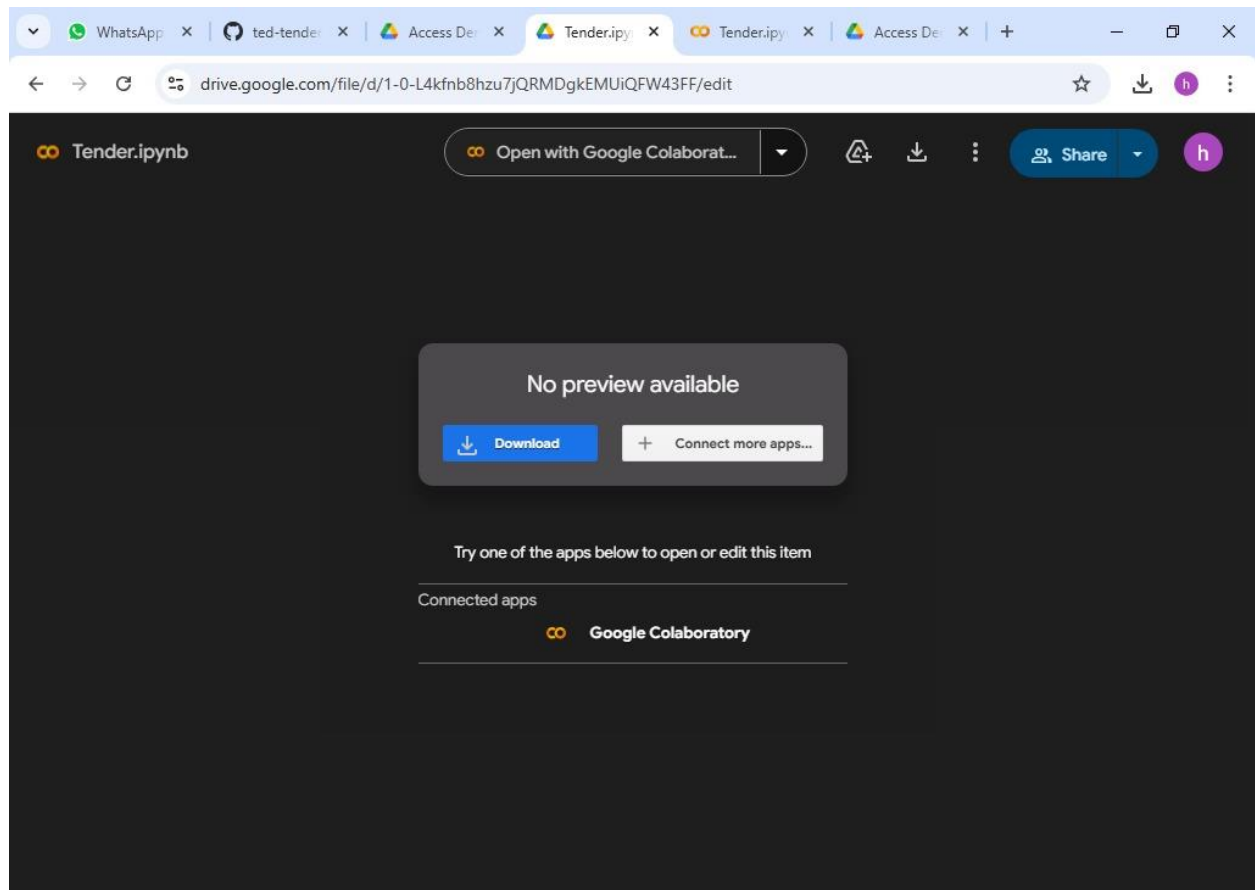
Request access



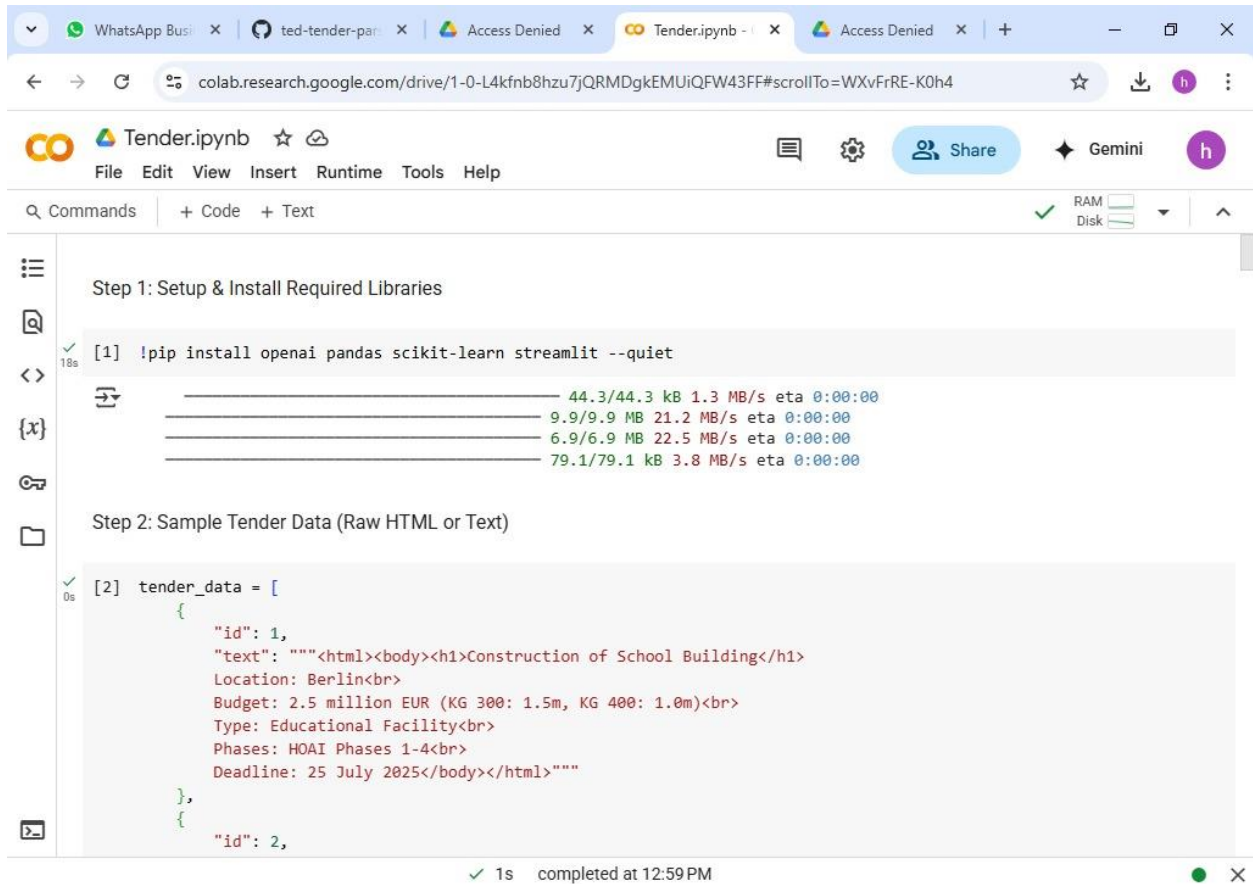
You're signed in as

 hamidjanjua83@gmail.com

Click the top and open google cloab directly



Then You will see an interface like this



The screenshot displays the Tender.ipynb interface, which is a web-based environment for running code. The interface includes a browser window at the top with the URL `colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4`. Below the browser window, the Tender.ipynb logo and navigation menu (File, Edit, View, Insert, Runtime, Tools, Help) are visible. The main area is divided into two sections: Step 1: Setup & Install Required Libraries and Step 2: Sample Tender Data (Raw HTML or Text).

Step 1: Setup & Install Required Libraries

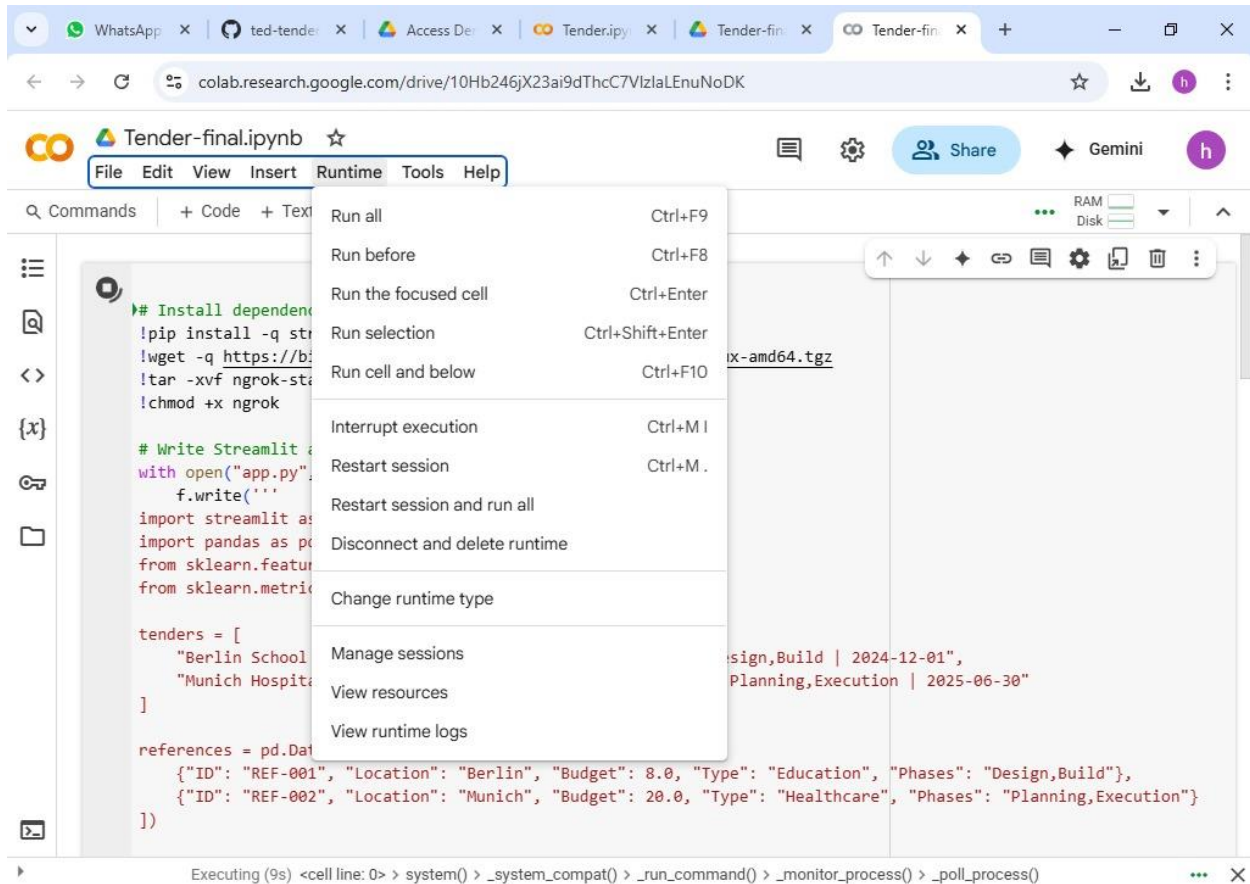
```
[1] !pip install openai pandas scikit-learn streamlit --quiet
```

Step 2: Sample Tender Data (Raw HTML or Text)

```
[2] tender_data = [
    {
        "id": 1,
        "text": """<html><body><h1>Construction of School Building</h1>
Location: Berlin<br>
Budget: 2.5 million EUR (KG 300: 1.5m, KG 400: 1.0m)<br>
Type: Educational Facility<br>
Phases: HOAI Phases 1-4<br>
Deadline: 25 July 2025</body></html>"""
    },
    {
        "id": 2,
```

The interface also shows a status bar at the bottom indicating that the code was completed at 12:59 PM.

Then click 'Run All' and it will start running



The screenshot shows a Google Colab notebook titled 'Tender-final.ipynb'. The 'Runtime' menu is open, displaying various options for running the code. The 'Run all' option is highlighted, which corresponds to the keyboard shortcut Ctrl+F9. The notebook content includes a code cell with the following text:

```
!# Install dependencies
!pip install -q streamlit
!wget -q https://bin.equinox.io/c/4VmDd16TlQH/ngrok-stable-linux-amd64.tgz
!tar -xvf ngrok-stable-linux-amd64.tgz
!chmod +x ngrok

# Write Streamlit app
with open("app.py", "w") as f:
    f.write("""
import streamlit as st
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

tenders = [
    {"ID": "REF-001", "Location": "Berlin", "Budget": 8.0, "Type": "Education", "Phases": "Design,Build"},
    {"ID": "REF-002", "Location": "Munich", "Budget": 20.0, "Type": "Healthcare", "Phases": "Planning,Execution"}
]

references = pd.DataFrame(tenders)

st.title("Tender Analysis")
st.write(references)
""")
```

The bottom status bar indicates the notebook is 'Executing (9s)' and shows the current command line: `<cell line: 0> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()`.

WhatsApp Busi x ted-tender-par x Access Denied x Tender.ipynb x Access Denied x

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4

Tender.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

79.1/79.1 KB 3.8 MB/s eta 0:00:00

RAM Disk

Step 2: Sample Tender Data (Raw HTML or Text)

```
[2] tender_data = [
    {
        "id": 1,
        "text": """<html><body><h1>Construction of School Building</h1>
        Location: Berlin<br>
        Budget: 2.5 million EUR (KG 300: 1.5m, KG 400: 1.0m)<br>
        Type: Educational Facility<br>
        Phases: HOAI Phases 1-4<br>
        Deadline: 25 July 2025</body></html>"""
    },
    {
        "id": 2,
        "text": """<html><body><h1>Hospital Renovation</h1>
        Location: Munich<br>
        Budget: 5.2 million EUR (KG 300: 3.0m, KG 400: 2.2m)<br>
        Type: Healthcare<br>
        Phases: HOAI 2-5<br>
        Deadline: 18 August 2025</body></html>"""
    },
    {
        "id": 3,
        "text": """<html><body><h1>New Office Complex</h1>
        Location: Hamburg<br>
        Budget: 8.1 million EUR (KG 300: 4.6m, KG 400: 3.5m)<br>
        Type: Commercial<br>
        Phases: HOAI Phases 1-7<br>
        Deadline: 5 September 2025</body></html>"""
    }
]
```

1s completed at 12:59PM

WhatsApp Busi x ted-tender-par x Access Denied x Tender.ipynb x Access Denied x

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4

Tender.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini

RAM Disk

Step 3: GPT Mock Parsing Function

0s

```
import re
import json

def parse_tender_html(html_text):
    def extract(pattern, text):
        match = re.search(pattern, text, re.IGNORECASE)
        return match.group(1).strip() if match else None

    return {
        "project_name": extract(r"<h1>(.*?)</h1>", html_text),
        "location": extract(r"Location:\s*(.*?)<br>", html_text),
        "budget": extract(r"Budget:\s*([\d\.]+)\s*million", html_text),
        "project_type": extract(r"Type:\s*(.*?)<br>", html_text),
        "phases": extract(r"Phases:\s*(.*?)<br>", html_text),
        "deadline": extract(r"Deadline:\s*(.*?)</body>", html_text)
    }

parsed_tenders = [parse_tender_html(item["text"]) for item in tender_data]

import pandas as pd
df_tenders = pd.DataFrame(parsed_tenders)
df_tenders
```

1s completed at 12:59 PM

WhatsApp Busi x ted-tender-par x Access Denied x Tender.ipynb x Access Denied x

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4

Tender.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini

```
import pandas as pd
df_tenders = pd.DataFrame(parsed_tenders)
df_tenders
```

1 to 3 of 3 entries

index	project_name	location	budget	project_type	phases	deadline
0	Construction of School Building	Berlin	2.5	Educational Facility	HOAI Phases 1-4	25 July 2025
1	Hospital Renovation	Munich	5.2	Healthcare	HOAI 2-5	18 August 2025
2	New Office Complex	Hamburg	8.1	Commercial	HOAI Phases 1-7	5 September 2025

Show 25 per page

Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

Categorical distributions

2-d categorical distributions

1s completed at 12:59 PM



WhatsApp Busi...ted-tender-par...Access Denied...Tender.ipynb...Access Denied...+

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4

CO Tender.ipynb ☆

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

✓ RAM Disk

Share Gemini h

⋮

🔍

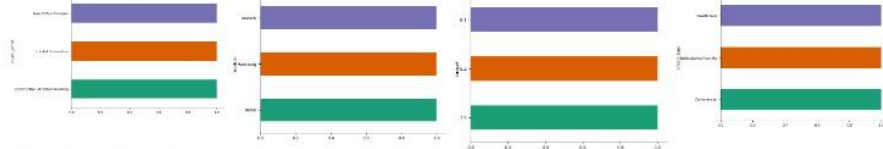
<>

{x}

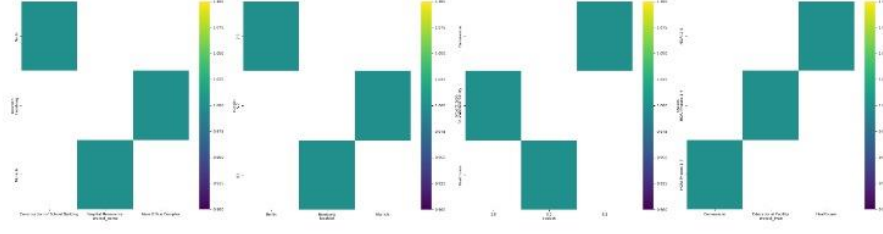
🔑

📁

### Categorical distributions



### 2-d categorical distributions



Next steps:

Generate code with df\_tenders

View recommended plots

New interactive sheet

📄

Step 4: Reference Data Load (Mock Excel Data)

✓ 1s completed at 12:59 PM

● ✕

WhatsApp Busi x ted-tender-par x Access Denied x Tender.ipynb x Access Denied x

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4

Tender.ipynb

File Edit View Insert Runtime Tools Help

Commands

+ Code + Text

Next steps:

Generate code with df\_tenders

View recommended plots

New interaction

RAM

Disk

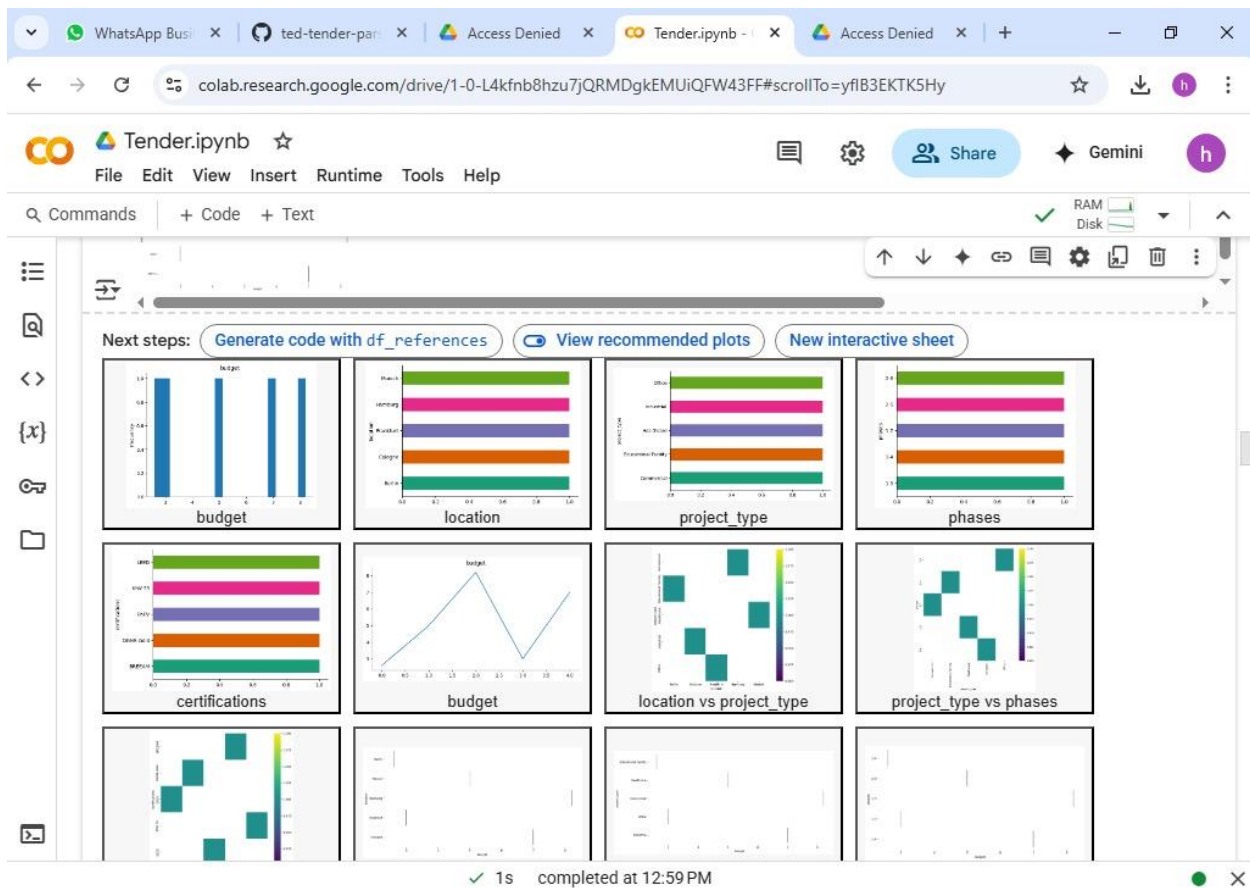
Step 4: Reference Data Load (Mock Excel Data)

[4] data = {  
    "location": ["Berlin", "Munich", "Hamburg", "Frankfurt", "Cologne"],  
    "budget": [2.6, 5.0, 8.2, 3.0, 7.0],  
    "project\_type": ["Educational Facility", "Healthcare", "Commercial", "Office", "Industrial"],  
    "phases": ["1-4", "2-5", "1-7", "1-3", "2-6"],  
    "certifications": ["DGNB Gold", "BREEAM", "LEED", "EnEV", "KfW 55"]  
}  
df\_references = pd.DataFrame(data)  
df\_references

	location	budget	project_type	phases	certifications
0	Berlin	2.6	Educational Facility	1-4	DGNB Gold
1	Munich	5.0	Healthcare	2-5	BREEAM
2	Hamburg	8.2	Commercial	1-7	LEED
3	Frankfurt	3.0	Office	1-3	EnEV
4	Cologne	7.0	Industrial	2-6	KfW 55

1s

completed at 12:59 PM



WhatsApp Busi x ted-tender-par x Access Denied x Tender.ipynb x Access Denied x

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=yflB3EKTk5Hy

Tender.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

RAM Disk

certifications

budget

location vs project\_type

project\_type vs phases

phases vs certifications

location vs budget

project\_type vs budget

phases vs budget

certifications vs budget

Step 5: Semantic Matching Function (Simple Rule-based)

✓ 2s

[5] from sklearn.metrics.pairwise import cosine\_similarity  
from sklearn.feature\_extraction.text import TfidfVectorizer

✓ 1s completed at 12:59 PM

WhatsApp Busi x ted-tender-par x Access Denied x Tender.ipynb - x Access Denied x +

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4

CO

Tender.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini h

Commands + Code + Text RAM Disk

certifications vs budget

Step 5: Semantic Matching Function (Simple Rule-based)

[5] ✓ 2s

```
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer

def match_references(tender, references_df):
    explanations = []
    scores = []

    for idx, ref in references_df.iterrows():
        score = 0
        explanation = []

        if tender["location"].lower() == ref["location"].lower():
            score += 1
            explanation.append("Same location")

        if abs(float(tender["budget"]) - ref["budget"]) <= 1.0:
            score += 1
            explanation.append("Budget close")

        if tender["project_type"].lower() == ref["project_type"].lower():
            score += 1
```

✓ 1s completed at 12:59 PM

WhatsApp Busi X ted-tender-par X Access Denied X Tender.ipynb X Access Denied X

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4

Tender.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini

```
if abs(float(tender["budget"]) - ref["budget"]) <= 1.0:
    score += 1
    explanation.append("Budget close")

if tender["project_type"].lower() == ref["project_type"].lower():
    score += 1
    explanation.append("Same type")

tender_phase = tender["phases"].replace("HOAI", "").strip().split("-")
ref_phase = ref["phases"].split("-")
if tender_phase[0] <= ref_phase[0] and tender_phase[-1] >= ref_phase[-1]:
    score += 1
    explanation.append("Phases compatible")

scores.append(score)
explanations.append(", ".join(explanation))

df_result = references_df.copy()
df_result["match_score"] = scores
df_result["explanation"] = explanations
df_result = df_result.sort_values(by="match_score", ascending=False).head(3)
return df_result

# Example: Match first tender
match_references(parsed_tenders[0], df_references)
```

1 to 3 of 3 entries Filter

✓ 1s completed at 12:59 PM

X

WhatsApp Busi x ted-tender-par x Access Denied x Tender.ipynb x Access Denied x +

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=WXvFrRE-K0h4

Tender.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini h

RAM Disk

Step 6: Match All Tenders with Top 3 References

```
[6] all_matches = []

for tender in parsed_tenders:
    match_df = match_references(tender, df_references)
    tender_result = {
        "project_name": tender["project_name"],
        "matches": match_df.to_dict(orient="records")
    }
    all_matches.append(tender_result)

# Display first tender match result
import pprint
pprint.pprint(all_matches[0])
```

```
{'matches': [{'budget': 2.6,
               'certifications': 'DGNB Gold',
               'explanation': 'Same location, Budget close, Same type',
               'location': 'Berlin',
               'match_score': 3,
               'phase': '1-A'}
```

1s completed at 12:59 PM





WhatsApp Busi x ted-tender-par x Access Denied x Tender.ipynb - x Access Denied x +

colab.research.google.com/drive/1-0-L4kfnb8hzu7jQRMDgkEMUiQFW43FF#scrollTo=AJ73I1wOLp6O

Tender.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini h

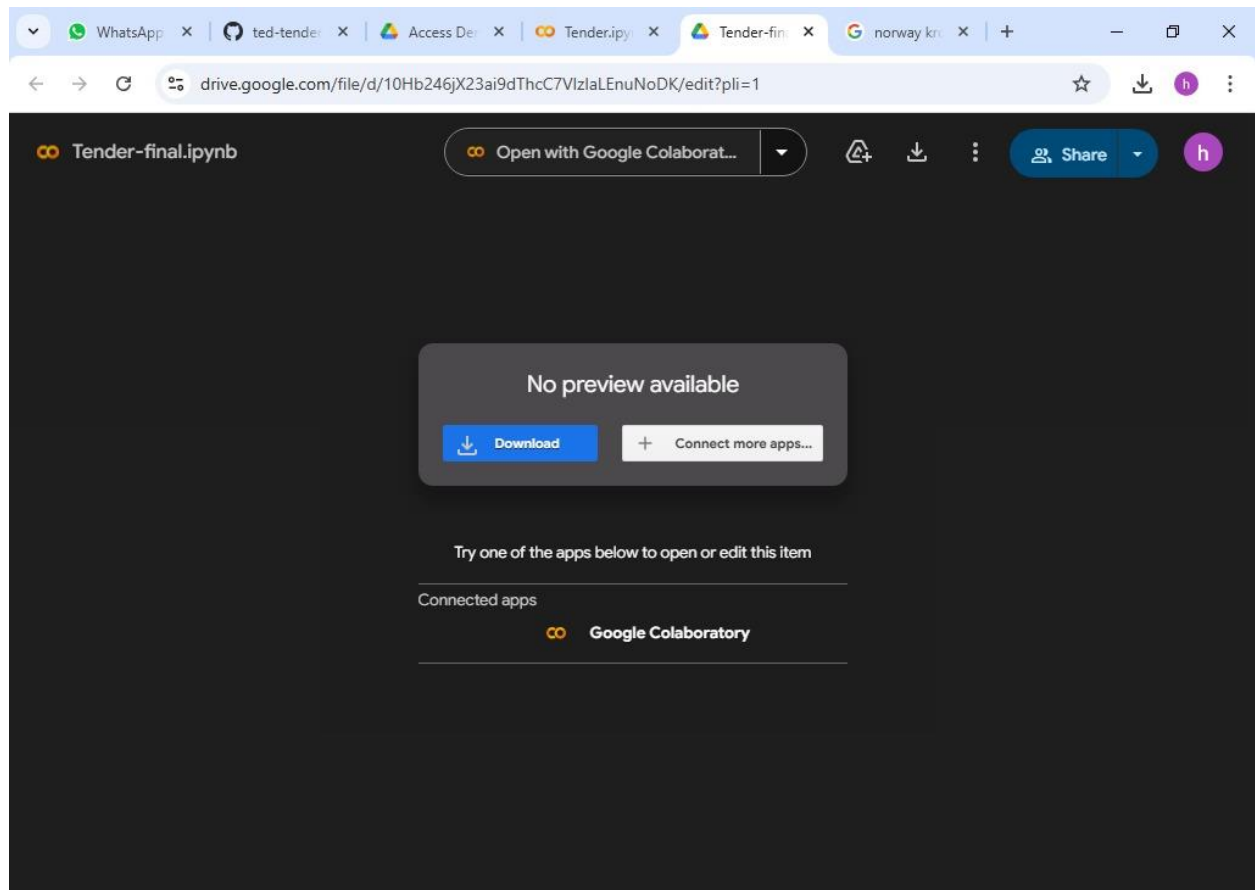
[10] time.sleep(1)

Step 9: Simulated Daily Automation (Extra Bonus)

[11] import schedule  
import time  
  
def daily\_run():  
 print("Running simulated tender parser...")  
 # You can reuse parse\_tender\_html() and match\_references() here  
 # Add more tenders or simulate new data in this block  
 print("Completed run.")  
  
 # Run once every day at 9 AM (simulation)  
 schedule.every().day.at("09:00").do(daily\_run)  
  
 # Simulate the scheduler loop (this is just a demo, won't actually run daily in Colab)  
 for \_ in range(2):  
 schedule.run\_pending()  
 time.sleep(1)


1s completed at 12:59 PM

Same you click this final google cloab click



WhatsApp x | ted-tende x | Access De x | Tender.ipyn x | Tender-fin x | Tender-fin x + - □ ×

colab.research.google.com/drive/10Hb246jX23ai9dThcC7VlzlaLEnuNoDK ☆ ⬇ h ⋮

 Tender-final.ipynb ☆ ☁

File Edit View Insert Runtime Tools Help

🔍 Commands | + Code + Text Connect ▾ ⬆

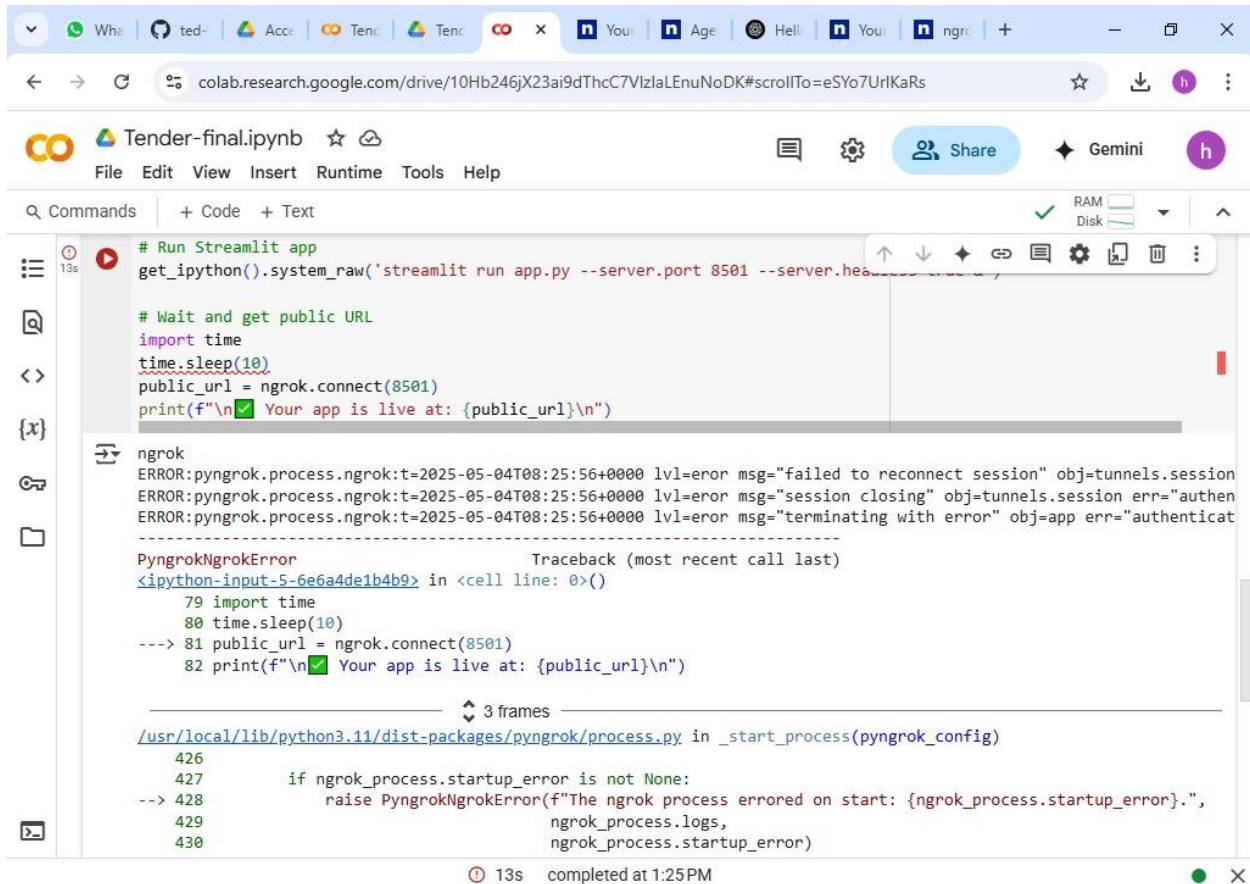
```
# Install dependencies
!pip install -q streamlit pyngrok pandas scikit-learn
!wget -q https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.tgz
!tar -xvf ngrok-stable-linux-amd64.tgz
!chmod +x ngrok

# Write Streamlit app to file
with open("app.py", "w") as f:
    f.write('''
import streamlit as st
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

tenders = [
    "Berlin School Construction | Berlin | 8.5 | Education | Design,Build | 2024-12-01",
    "Munich Hospital Renovation | Munich | 22.0 | Healthcare | Planning,Execution | 2025-06-30"
]

references = pd.DataFrame([
    {"ID": "REF-001", "Location": "Berlin", "Budget": 8.0, "Type": "Education", "Phases": "Design,Build"},
    {"ID": "REF-002", "Location": "Munich", "Budget": 20.0, "Type": "Healthcare", "Phases": "Planning,Execution"}
])
```

But you will get an error because for that, you'll need to create an account on Ngrok, as the free version only allows one agent



The screenshot shows a Google Colab notebook titled "Tender-final.ipynb". The code in the notebook is as follows:

```
# Run Streamlit app
get_ipython().system_raw('streamlit run app.py --server.port 8501 --server.headless=True')

# Wait and get public URL
import time
time.sleep(10)
public_url = ngrok.connect(8501)
print(f"\n✅ Your app is live at: {public_url}\n")
```

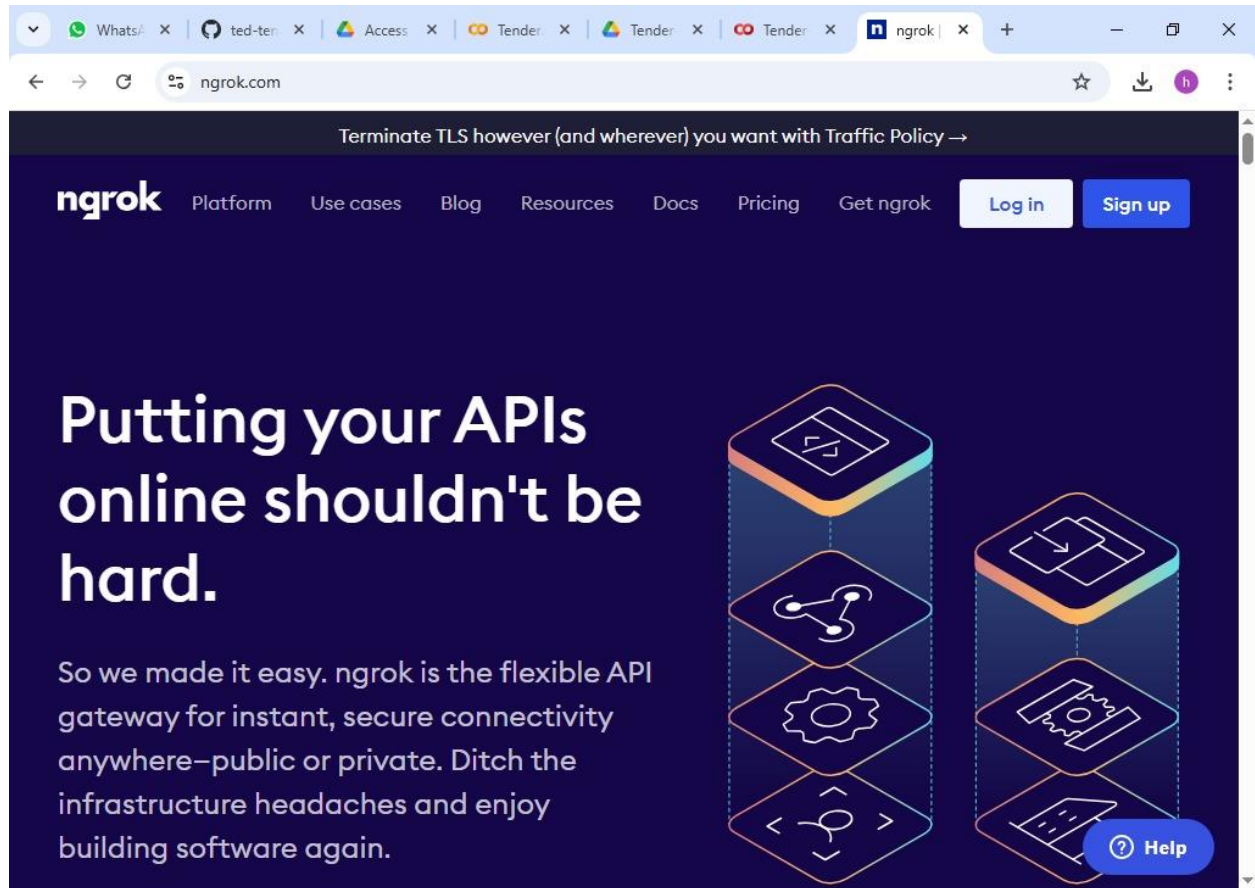
The output of the code shows several error messages from the ngrok process:

```
ngrok
ERROR:pyngrok.process.ngrok:t=2025-05-04T08:25:56+0000 lvl=error msg="failed to reconnect session" obj=tunnels.session
ERROR:pyngrok.process.ngrok:t=2025-05-04T08:25:56+0000 lvl=error msg="session closing" obj=tunnels.session err="authen
ERROR:pyngrok.process.ngrok:t=2025-05-04T08:25:56+0000 lvl=error msg="terminating with error" obj=app err="authenticat
-----
PyngrokNgrokError                                Traceback (most recent call last)
<ipython-input-5-6e6a4de1b4b9> in <cell line: 0>()
    79 import time
    80 time.sleep(10)
--> 81 public_url = ngrok.connect(8501)
    82 print(f"\n✅ Your app is live at: {public_url}\n")

-----
3 frames
/usr/local/lib/python3.11/dist-packages/pyngrok/process.py in _start_process(pyngrok_config)
    426
    427     if ngrok_process.startup_error is not None:
--> 428         raise PyngrokNgrokError(f"The ngrok process errored on start: {ngrok_process.startup_error}.",
    429                                 ngrok_process.logs,
    430                                 ngrok_process.startup_error)
```

The notebook interface shows a status bar at the bottom indicating "13s completed at 1:25PM".

Go to this website and create account simple click sign up





## Log in

Email

zulqarnainfuast9@gmail.com

Password

.....

Log in

[I forgot my password](#)

or

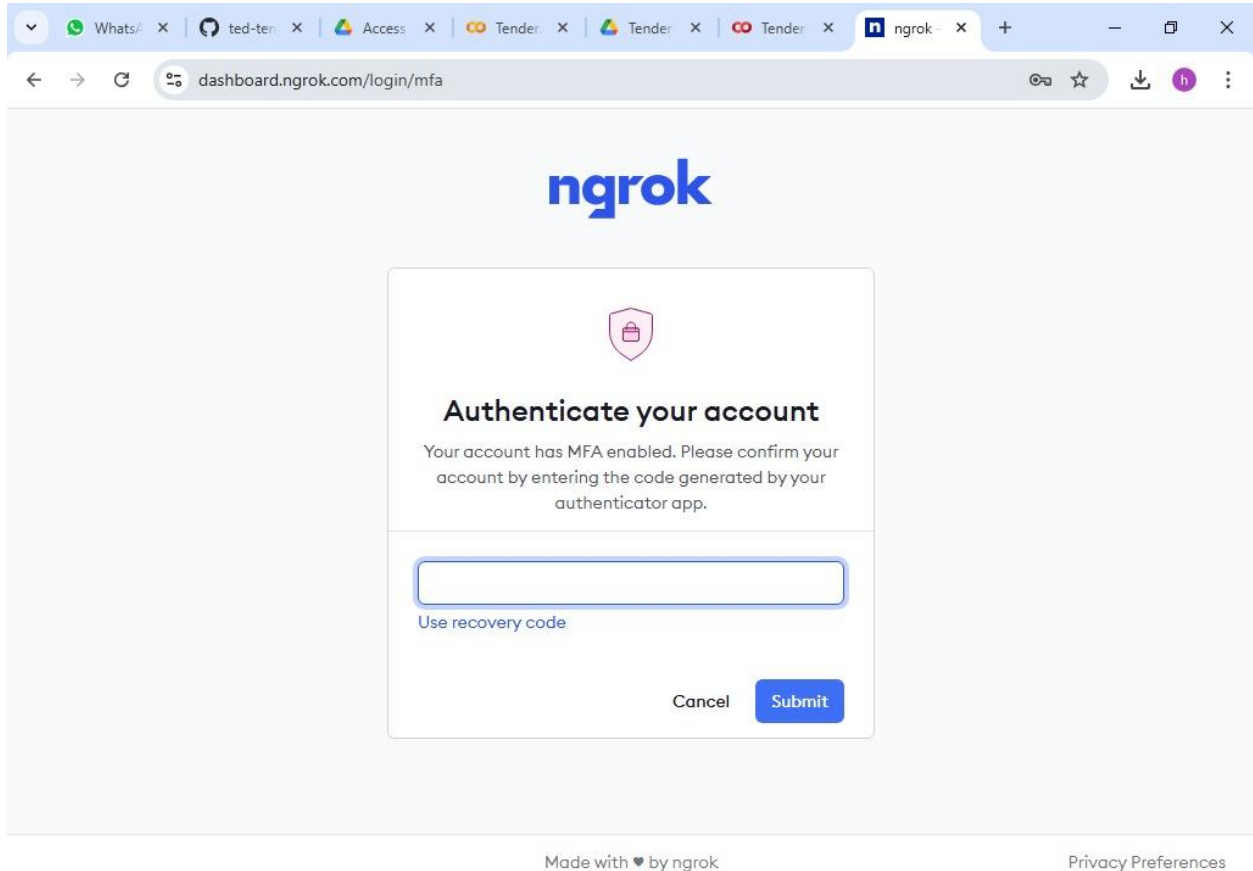
 Log in with GitHub

 Log in with Google

Made with ♥ by ngrok


[Privacy Preferences](#)

Download the Authenticator app from the Play Store and verify your account email



The screenshot shows a web browser window with the address bar displaying `dashboard.ngrok.com/login/mfa`. The page features the ngrok logo at the top center. Below the logo is a white box containing a shield icon with a keyhole. The text inside the box reads: "Authenticate your account", "Your account has MFA enabled. Please confirm your account by entering the code generated by your authenticator app.", a text input field, a link "Use recovery code", and two buttons: "Cancel" and "Submit". At the bottom of the page, it says "Made with ♥ by ngrok" and "Privacy Preferences".

**ngrok**



**Authenticate your account**

Your account has MFA enabled. Please confirm your account by entering the code generated by your authenticator app.

[Use recovery code](#)

[Cancel](#) [Submit](#)

Made with ♥ by ngrok [Privacy Preferences](#)

✓ Logged in! Redirecting...



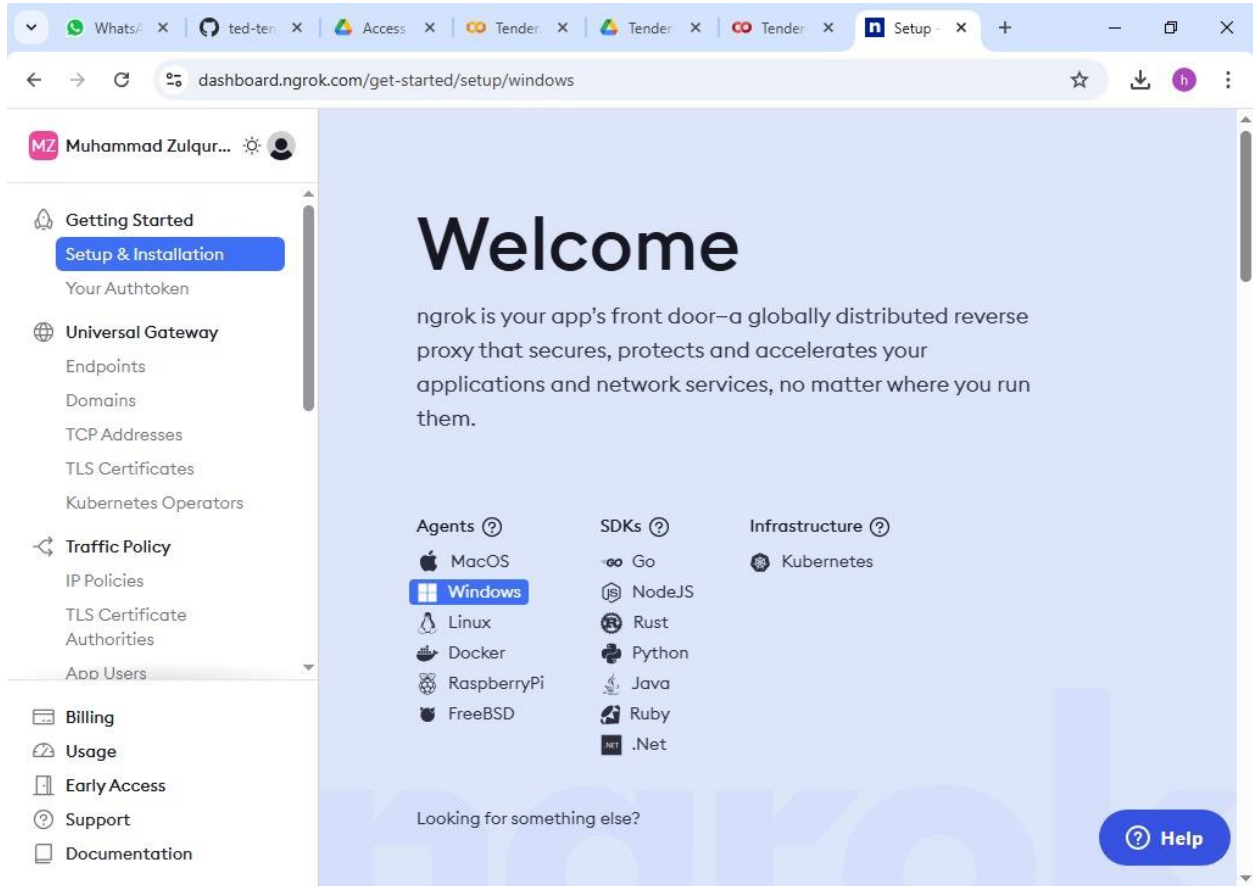
### Authenticate your account

Please confirm your account by entering the code generated by your authenticator app.

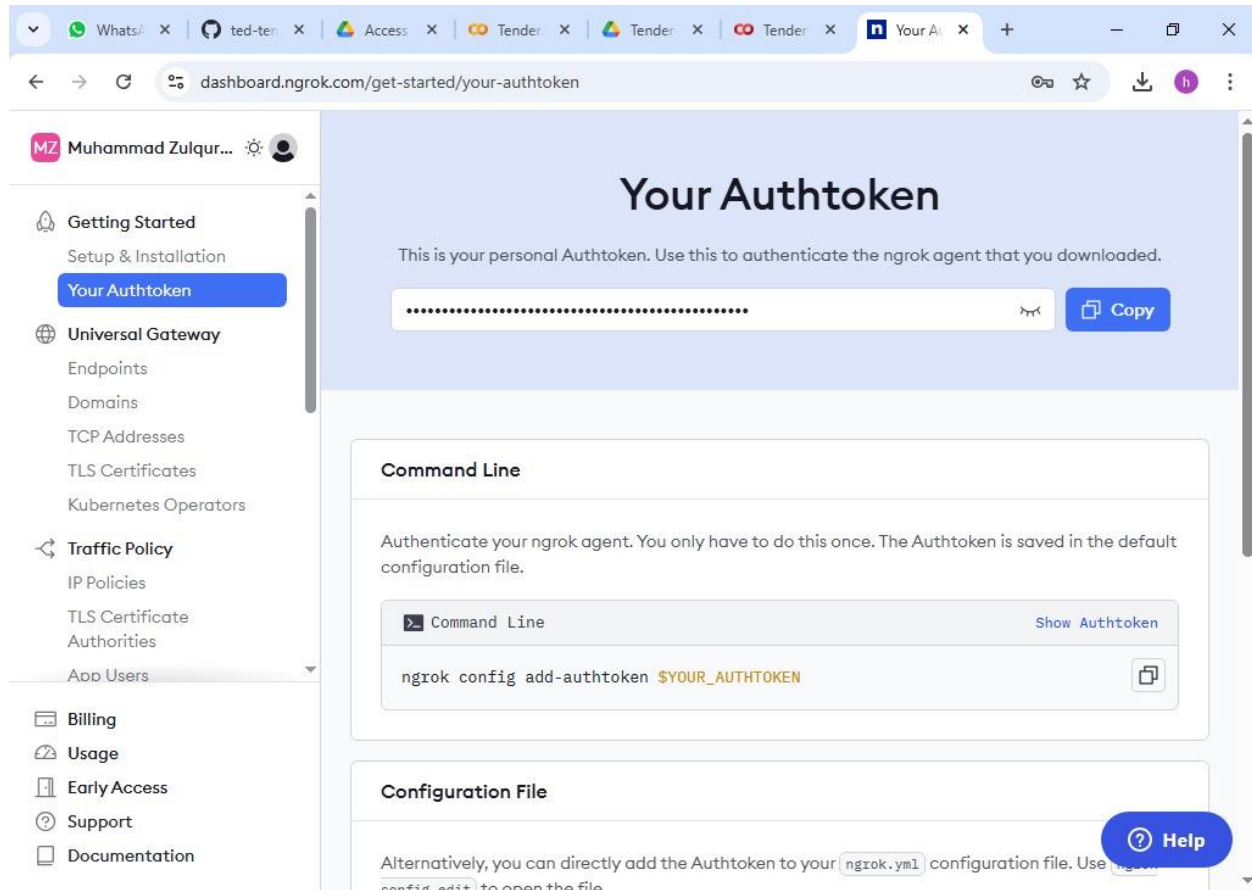
Verified!



After verify you see this interface and click “Your Authtoken”



Copy this token and replace it with my token



The screenshot shows the Ngrok dashboard in a web browser. The browser's address bar displays the URL `dashboard.ngrok.com/get-started/your-authtoken`. The dashboard has a light blue header with the title "Your Authtoken". Below the title, a message states: "This is your personal Authtoken. Use this to authenticate the ngrok agent that you downloaded." A text input field contains a masked token (dots), and a blue "Copy" button is to its right. The left sidebar contains a navigation menu with categories: "Getting Started" (Setup & Installation, Your Authtoken), "Universal Gateway" (Endpoints, Domains, TCP Addresses, TLS Certificates, Kubernetes Operators), "Traffic Policy" (IP Policies, TLS Certificate Authorities, Add Users), "Billing", "Usage", "Early Access", "Support", and "Documentation". The "Your Authtoken" item is highlighted. Below the main token display, there are two sections: "Command Line" and "Configuration File". The "Command Line" section includes the instruction "Authenticate your ngrok agent. You only have to do this once. The Authtoken is saved in the default configuration file." and a code block with the command `ngrok config add-authtoken $YOUR_AUTHTOKEN`. The "Configuration File" section starts with the text "Alternatively, you can directly add the Authtoken to your `ngrok.yml` configuration file. Use `ngrok config edit` to open the file". A blue "Help" button is located in the bottom right corner of the dashboard.

**Your Authtoken**

This is your personal Authtoken. Use this to authenticate the ngrok agent that you downloaded.

..... [Copy](#)

**Command Line**

Authenticate your ngrok agent. You only have to do this once. The Authtoken is saved in the default configuration file.

`ngrok config add-authtoken $YOUR_AUTHTOKEN` [Show Authtoken](#)

**Configuration File**

Alternatively, you can directly add the Authtoken to your `ngrok.yml` configuration file. Use `ngrok config edit` to open the file

[Help](#)

MZ Muhammad Zulqur...

Getting Started

Setup & Installation

Your Authtoken

Universal Gateway

Endpoints

Domains

TCP Addresses

TLS Certificates

Kubernetes Operators

Traffic Policy

IP Policies

TLS Certificate Authorities

App Users

Billing

Usage

Early Access

Support

Documentation

# Your Authtoken

This is your personal Authtoken. Use this to authenticate the ngrok agent that you downloaded.

2wcSpEOIQIKIe502vCrYxYYESCw\_py7haPfEu22LbFrZ6URB

Copy

## Command Line

Authenticate your ngrok agent. You only have to do this once. The Authtoken is saved in the default configuration file.

Command Line

Hide Authtoken

```
ngrok config add-authtoken 2wcSpEOIQIKIe502vCrYxYYESCw_py7haPfEu22LbFrZ6URB
```

## Configuration File

Alternatively, you can directly add the Authtoken to your `ngrok.yml` configuration file. Use `ngrok`

Help

Paste it here

The screenshot shows a Google Colab notebook interface. The browser address bar indicates the notebook is located at [colab.research.google.com/drive/10Hb246jX23ai9dThcC7VlzlaLEnuNoDK#scrollTo=eSYo7UrKaRs](https://colab.research.google.com/drive/10Hb246jX23ai9dThcC7VlzlaLEnuNoDK#scrollTo=eSYo7UrKaRs). The notebook title is "Tender-final.ipynb". The code in the cell is as follows:

```
st.success("Exported!")

if __name__ == "__main__":
    main()
...

# Kill previous processes if running
import os
os.system('pkill -f streamlit')
os.system('pkill -f ngrok')

# Setup ngrok
from pyngrok import ngrok
ngrok.set_auth_token("12wcimk6EoWl0NsnHkVc5kv7a4zm_3C2n9hJou69pYGrPMPm2q")

# Run Streamlit app
get_ipython().system_raw('streamlit run app.py --server.port 8501 --server.headless true &')

# Wait and get public URL
import time
time.sleep(10)
public_url = ngrok.connect(8501)
print(f"\n✅ Your app is live at: {public_url}\n")
```

The output of the code execution is shown in the bottom panel:

```
ngrok
✅ Your app is live at: NgrokTunnel: "https://8b56-34-86-192-149.ngrok-free.app" -> "http://localhost:8501"
```

The execution status at the bottom indicates it took 13 seconds and was completed at 1:32 PM.

What's New

ted-t

Acces

Tende

Tende

Tende

Your

Intro

Hello

n A x

+

—

□

×

← → ↺ dashboard.ngrok.com/agents ☆ ⬇ h ⋮

MZ Muhammad Zulqur... ⚙️ 👤

🔔 Getting Started

Setup & Installation

Your Authtoken

🌐 Universal Gateway

Endpoints

Domains

TCP Addresses

TLS Certificates

Kubernetes Operators

↔ Traffic Policy

IP Policies

TLS Certificate Authorities

App Users

📄 Billing

🕒 Usage

📄 Early Access

🔍 Support

📖 Documentation

# Agents

Instances of running ngrok agents. You can stop, restart, or update agents from this page.

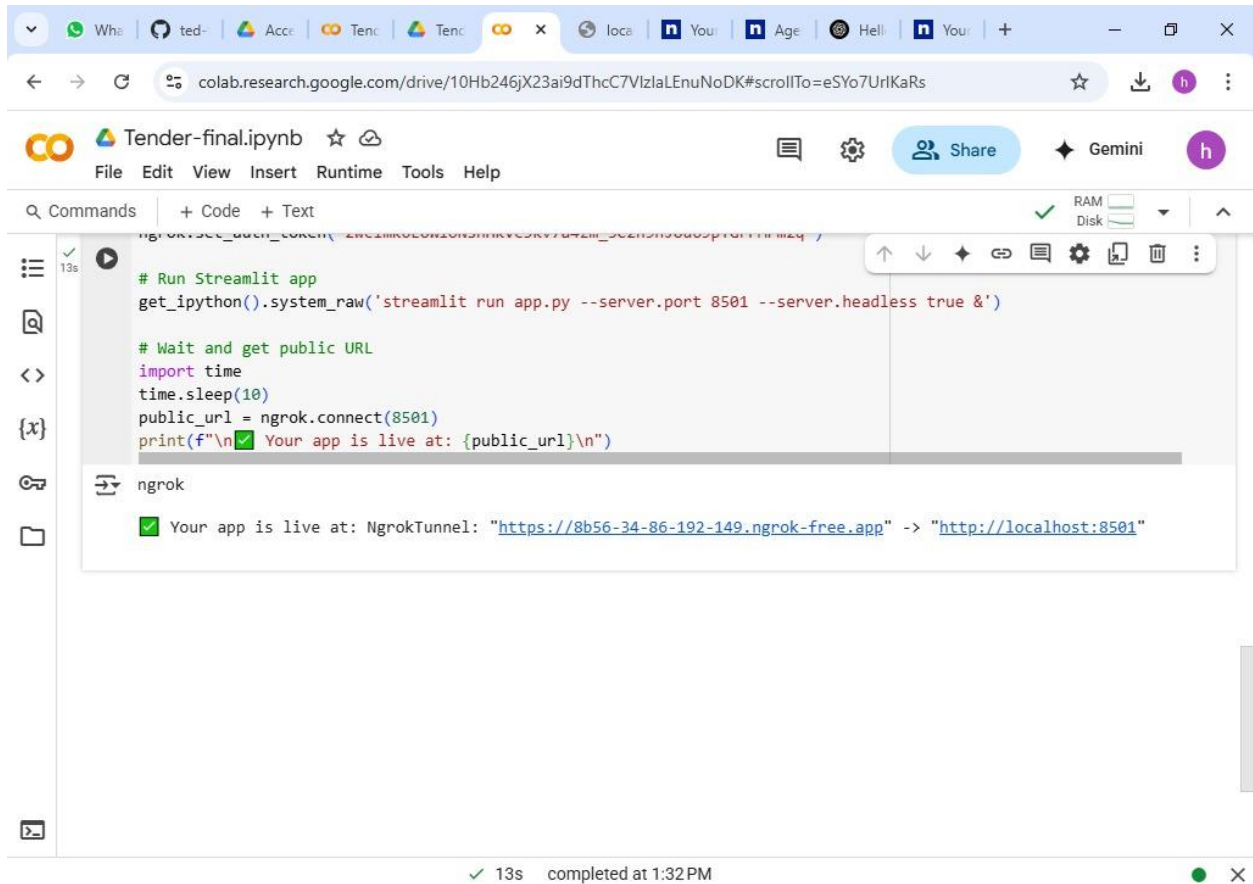
API Docs

ID ↕	Region ↕	IP ↕	Agent Version ↕	Tunnels Online ↕	Started By	
ts...3uR6Qj	US	35.231.142.177	ngrok-agent/3.22.1	1 Online	Muhammad Zulquznain	

?

 Help

If this link is shown, it means the project is running



The screenshot shows a Google Colab notebook titled "Tender-final.ipynb". The code cell contains the following Python code:

```
# Run Streamlit app
get_ipython().system_raw('streamlit run app.py --server.port 8501 --server.headless true &')

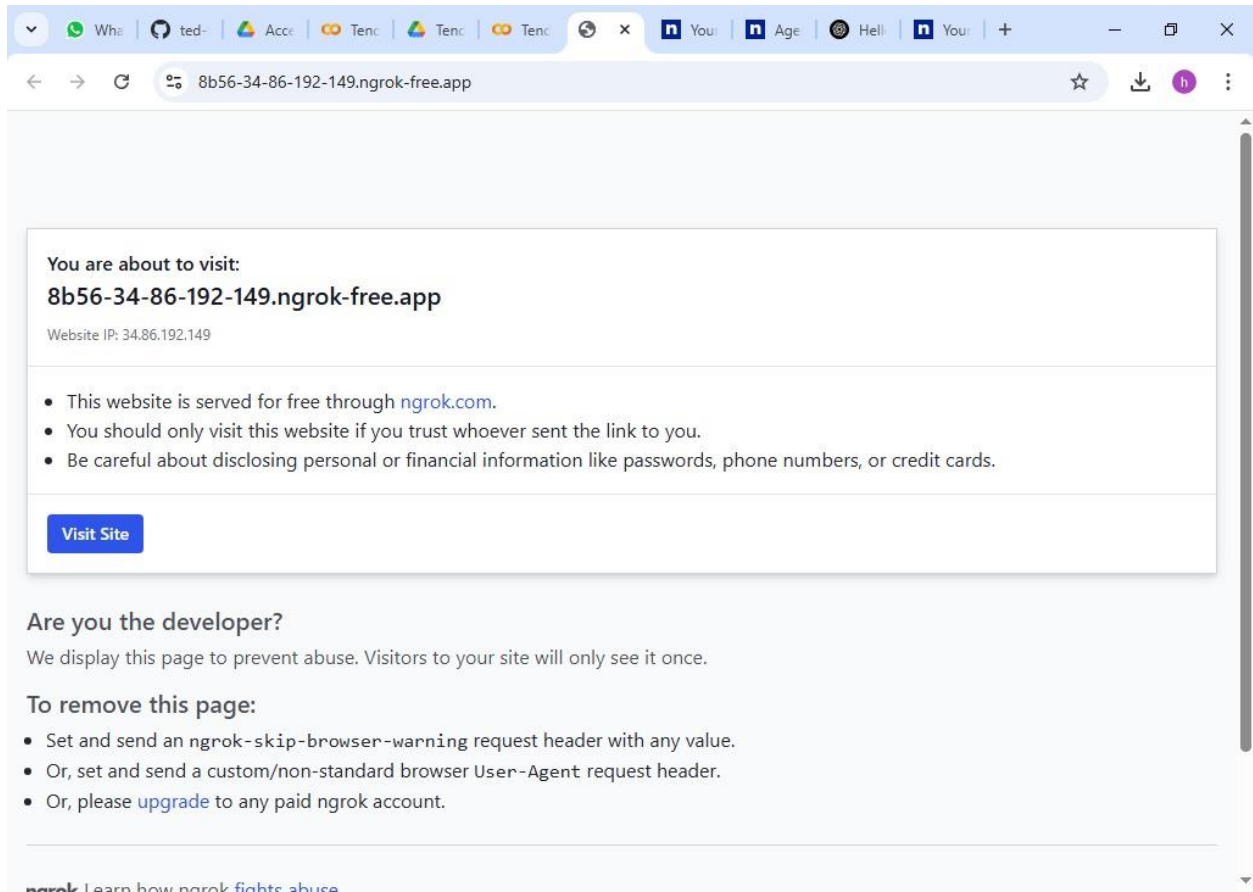
# Wait and get public URL
import time
time.sleep(10)
public_url = ngrok.connect(8501)
print(f"\n✅ Your app is live at: {public_url}\n")
```

The output of the code cell shows the public URL:

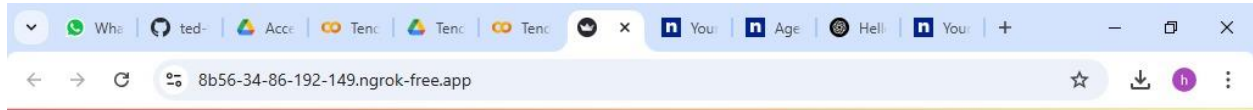
```
ngrok
✅ Your app is live at: NgrokTunnel: "https://8b56-34-86-192-149.ngrok-free.app" -> "http://localhost:8501"
```

The status bar at the bottom indicates that the code was executed successfully, taking 13 seconds, and completed at 1:32 PM.

Then click visit site



The project is live now; you can see everything and also download the CSV file



## TED Tender Parser MVP

Select Tender

Berlin School Construction | Berlin | 8.5 | Education | Design,Build | 2024-12-01

Find Matches







# TED Tender Parser MVP

Select Tender

Munich Hospital Renovation | Munich | 22.0 | Healthcare | Planning,Execution | 2025-06-30

Find Matches

## Parsed Tender

```
{
  "Project": "Munich Hospital Renovation"
  "Location": "Munich"
  "Budget": 22
  "Type": "Healthcare"
  "Phases": "Planning, Execution"
  "Deadline": "2025-06-30"
}
```

Export CSV

## Top 3 Matches

	ID	Location	Budget	Type	Phases
1	REF-002	Munich	20	Healthcare	Planning,Execution
0	REF-001	Berlin	8	Education	Design,Build

